A Cloud Comparison: AWS vs Google Cloud

Based on Machine Learning

Dhruvin Patel
University of Mississippi
Oxford, Mississippi
drpatel@go.olemiss.edu

ABSTRACT

Cloud computing and Machine Learning are some of the most trending words and topics in Computer Science. So much so that companies like Amazon and Google Cloud have been offering services of people to train and deploy Machine Learning models in the cloud. This paper aims to compare the instance types between Amazon's AWS and Google's Cloud based on Machine Learning training.

KEYWORDS

Cloud Computing, Machine Learning, Performance

ACM Reference Format:
Dhruvin Patel. 2018. A Cloud Comparison: AWS vs Google Cloud: Based on Machine Learning. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Cloud Computing and Machine Learning are two of the most growing topics in computer science. While these topics are not new, they have emerged with advantages in hardware, algorithms, and technology. Cloud Computing has become a resource that almost anyone can use and with Machine Learning popularity on the rise, the cloud can provide the hardware and resources to train and deploy models. Many devices and products rely on both Machine Learning and Cloud Computing. For example, the Internet of Things (IoTs) can use the cloud to send and receive data while also using Machine Learning models at are deployed on the cloud to get result and analysis the data. Companies like Amazon and Google provide instances to train Machine Learning models.

Cloud Instances provide users with the ability to connect to a virtual instance of a server and use the hardware and resources. Since training models can take a large amount of computing power to train the weights or classify, it can be a good comparison for different instances. Each cloud instance type uses different type of resources. Amazon's EC2 has multiple different type of instance based on the uses for the instance. Google's Cloud lets a user configure an instance to the desired resources they need. Also, the cost of instances varying between companies, knowing which instances are better of Machine Learning can save users time and money.

One of the new aspects of Cloud Computing is the illusion of infinite computing resources [7]. Can the cloud scale with machine learning models? Using multiple instances, a user can take advantage of the illusion of infinite computing resources. Amazon and Google both provide services that can use distributed TensorFlow to train models with multiple instances. Since training models can be done in parallel, it can use multiple instances. Having multiple instances will cut down the time it takes to train models and allow users to get trained models without upfront investment on hardware.

In this paper, it will present a comparison based on instance, cost analysis, and usability for Amazon and Google Cloud. Each company has the advantages and disadvantages. Also, the paper will present how to use distributed TensorFlow with both Amazon's SageMaker and Google's ML Cloud and a comparison between the two based on performance and user usability. The challenge is that Amazon and Google have different user interfaces and protocol for setting up a machine learning environment.

2 RELATED WORK

There have been studies into cloud benchmarking, but none I have found tests against two companies. For a comprehensive study of how to benchmark cloud product for scientific applications refer to [12]. Technology is ever changing, and cloud application have made advantages to become better at these applications. Instead of Varghese , six steps to benchmarking VMs, I will take a simpler to just use the training time of a simple model.

An article from D. Pop [10] discusses and surveys distributed solutions of Machine Learning with Cloud Computing. In this article he discusses solutions like using Apache Mahout, GraphLab, DrayadLINQ, and more. Since the paper was published in 2012, Google's TensorFlow has become a popular library of Machine Learning, and it does the ability to be coded a model for training on both local and distributed computing.

3 TESTING METHODOLOGY

Both Amazon and Google provide VM's for computing uses; however, the hardware on each instance type varies for each company. Amazon has different tiers of instances uses for example, T2 and M4 are listed for general purpose. In Table 1, it displays the instances name and the hardware resources it had that were tested on Amazon. Google cloud allows for more flexible where a user can either use choose a predefined instance's resources or choose the amount of CPU cores, number of GB of ram, and more. Since

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1: Instance Types on Amazon

vCPUs	Memory (GB)	Instance Name
1	2	t2.small
2	4	t2.medium
2	8	m4.large
4	16	m4.xlarge

Google Compute Engine allows this flexible, we can take advantage of this to compute to Amazon's EC2 instances.

In order to have similar results, the Convolution Natural Network (CNN) are the exact same of all instances. CNNs are used to classify images by subsampling layers to find features. The CNN will be training on is a dataset call MNIST which is a set of handwritten digits images. The network will train on the same number of steps and batch sizes of the training set. The timer will start before training the network and stop after the training is done. The model with be using Google's TensorFlow library [1].

For scaling on instances, Amazon and Google have products that can train Machine Learning networks with multiple instances. In Amazon's SageMaker, the user has to state how many instances and what type of instances to use. For this comparison, I choose to test on four ml.c5.2xlarge, which has 8 vCPUs and 16 GB RAM. On Google's CloudML, the user has to state what tier they want the model to be trained under. Google will scale the number of instances based on the job. For these comparison, I have set the Google's tier as Standard-1. Google states that the master is n1-highcpu-8 and the 4 worker instances with n1-highcpu-8. n1-highcpu-8 was 8 vCPU and 7.2 GB RAM [4]. During training, Google's CloudML used 4 workers instances. I choose these configurations so that Google and Amazons instances would use the same number of vCPUs.

Also in order to use multiple instances with TensorFlow, the data set has be converted to TFRecords. TFRecords allow no random access, therefore can be used for many instances to access the file [6]. The MINST dataset will be converted into a TFRecord file and stored in a bucket within the cloud. These will allow instances in both Amazon and Google's Cloud instances to have access to the record. In addition to preprocessing the dataset, Amazon requires to import a platform library to the code.

4 RESULTS

For an overall comparison, the section is broken down to three parts: Performance, Cost, and Usability. Performance will discuss the time it takes for each instance on each companys' compute engines. Cost will compare the cost for training on each instance. Usability will examine which company is better user friendly and easier to use.

4.1 Performance

Performance for this comparison is measured in the minutes an instance takes to train from the dataset. Using Table 1, to cross reference Amazon's instances types, Amazon's t2.small instances takes roughly 448 mins to train while on Google's instances takes

[1] All sources code was from example code for Google and Amazon with a few changes to make similar [8, 9, 11]

Figure 1: Training Time in Minutes per Instance

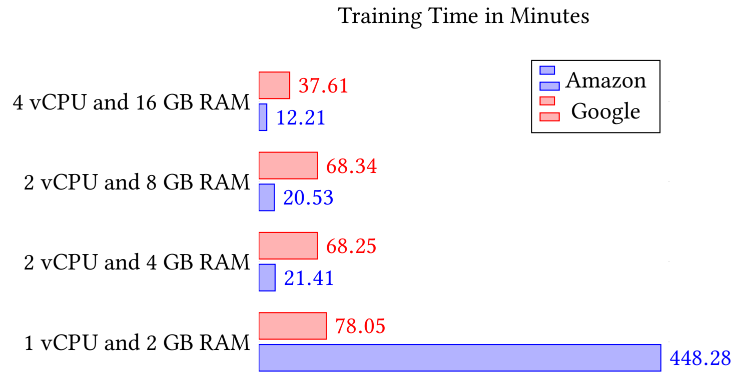
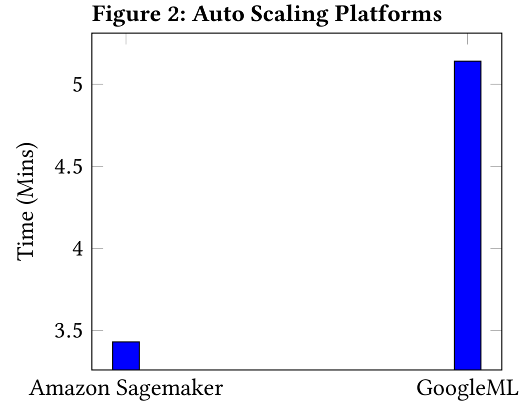


Figure 2: Auto Scaling Platforms



approximately 78 mins. Google is over 5 times faster than Amazon instance with the same resources. This is because Amazon's t2 instances are classify as Burstable Performance Instances [1]. This means the instances uses CPU Credit per hours. So, if an instance uses all of its CPU Credits, all of this process will have a baseline performance. For example, in t2.small the baseline is 10 percent of CPU utilization. Therefore, when all the instance cpu credits are used, the instance will only use 10 percent of the CPU power. I believe these instances are more attended for website host where the demand of the CPU is not consistent using 100 percent of the CPU.

In Figure 1, For running instance with 2 vCPUs and 4 GB RAM, Amazon instance finish training in 21.41 minutes while Google's instance took 68.25. Surprisingly, Amazon's instance is three times faster than Google's with the same resources. Similarly results for 2 vCPU and 8 GB RAM, where Amazon's instance finished in 20.53 mins and Google's instance finished in 68.34. This shows the amount of ram does not affect the training time of network. In the 4 vCPU and 16 GB RAM instances, Amazon finished training in 12.21 mins and Google finished in 37.21 mins. For these results, Amazon has clear advantage when it has more than one virtual CPU core.

Results from training Amazon and Googles scalable machine learning platforms were harder to compare since I could not match

Figure 3: GoogleML Workers CPU Utilization

Table 2: Instance Cost between Amazon and Google per Hour

vCPUs	Memory (GB)	Amazon	Google
1	2	$0.023	$0.030
2	4	$0.0464	$0.036
2	8	$0.1	$0.054
4	16	$0.2	$0.14

the amount of RAM. Moreover, in Figure 2 using four instances for Amazon’s SageMaker, it finished in 3.43 mins while Google’s CloudML finished in 5.14 seconds. In order to get a close comparison, the four instances for Amazon each had 8 vCPUs and Google’s instances had 8 vCPU per each instance. Each totaling to have 32 vCPUs but varying in RAM as mentioned in the testing methodology. From these results we can see that Amazon is faster than Google’s CloudML in training with multiple instances.

In Figure 3, Google provides metrics on the CPU utilization while training the network. The most all four worker instances reached was 73 percent at peak. All the worker instances start around 20-40 percent usage and increase based on the number of steps to train the model. All of Google’s worker instance seem to scale at the same amount until the training is done. Unfortunately, Amazon’s SageMaker does not give a chart with the CPU utilization. It would be a good comparison point to see how fast and when each instance scales depending on CPU utilization.

4.2 Cost

Depending on how large a dataset is and how complex a model is the longer it will take to train the model. The longer it takes to train a model the more it will cost. For most cloud computing providers, they charge per hour per instance.

In Table 2, I have summarized the cost between Amazon and Google instance for each instance I tested. The costs are in terms of per hour. From the table, Amazon is cheaper than Google when they both have 1 vCPU and 2 GB Memory [2]. However, has the amount of resources increase the cost of Amazon becomes more than Google’s instances. While most of Google instances cost less than Amazon, the cost to performance is better for Amazon since in most cases the Amazon instance finish training three times faster than Google’s Instances.

For training in multiple instances, Amazon’s SageMaker cost $1.904 per hour of 4 ml.c5.2xlarge instances and Google’s CloudML cost $2.9025 per hour of their predefined scale tier STANDARD-1 [3, 5]. From the pricing, Amazon is a significant amount cheaper than Google. Using multiple instance on Amazon is cheaper and performs quicker that training Machine Learning models.

4.3 Usability

Usability of this report is the ability of a user to set up an instance and start training a model. I found that Amazon’s approach to choosing an Amazon Machine Image (AMI) is much easier than installing all the libraries and packages need for training the model. Amazon has a deep learning AMI where TensorFlow, PyTorch, Keras, and more pre-installed. This saves the user from having to install the libraries and packages and also time. After choosing an AMI, Amazon creates a public key in order to ssh in to the instance. Then a user can either code the model in the instance shell or clone the code from GitHub.

In contrast, Google’s instances do not have an option to have pre-installed library or packages. The user will have to install each library. For this report in order to run the code, the library that need to be install are: Git, Python, and TensorFlow. However, connecting to the instance is easier of Google than Amazon. To connect to the instance shell, the user can just use the browser to open up a virtual shell. This eliminates the need for an SSH client or terminals.

Using Distributed TensorFlow makes training on multiple instance fairly simple. One advantage is using the same code for the model in one instance to multiple instances. For Amazon’s SageMaker, a user can create a notebook instance which is on a t2.medium and insert code for the estimator model and setup for the dataset. Then the user will have to preprocess the dataset to TFRecords. When a user wants to train the model, the user can call sagemaker.tensorflow class and set the number for training steps, evaluation steps, number of instances, type of instance, and more. This will create a job with in Amazon’s SageMaker. To set up in Google’s CloudML is very similar to Amazon’s SageMaker. One difference is that in Google’s Cloud files have to be structured in a particular way so that when a user creates a job in the Google shell, it can locate the files.

5 FUTURE WORK AND CONCLUSTION

This report does not include all of the instances Amazon and Google provides. For future work, I will need to test no more instances include GPU instances. GPU can make training model significantly faster do to the number of cores. Do to the free tier limitation for Amazon, I was not able to test the performance on these instances. In addition to GPU instances, testing multiple instances for Distributed TensorFlow can give a better comparison between the Machine Learning platforms for Amazon and Google. Using multiple GPU instances can dramatically improve the speed for training large datasets.

All in all, Amazon and Google both provide great ways to train Machine Learning models in the cloud. Amazon overall has better performance besides the lower end instances. Google’s cloud is slightly cheaper than Amazons instances. However, when it comes to Google and Amazon’s Machine Learning products like Cloud

ML and SageMaker, Amazon's Sagemaker is cheaper than Google. Also, Amazon makes it easier for user with their AMI instances, which lets the user not worry about installing different libraries. The raise of Cloud Computing and Machine Learning are hand to hand and comparing two companies to see which one performs better, cost less, and user friendly will save time and money.

REFERENCES

[1] [n. d.]. Amazon EC2 Instance Types. https://aws.amazon.com/ec2/instance-types/
[2] [n. d.]. Amazon EC2 Pricing. https://aws.amazon.com/ec2/pricing/on-demand/
[3] [n. d.]. Cloud ML Engine Pricing. https://cloud.google.com/ml-engine/docs/pricing
[4] [n. d.]. Cloud ML Enging Scale Tier. https://cloud.google.com/ml-engine/docs/tensorflow/training-overview/
[5] [n. d.]. Sagemaker Pricing. https://aws.amazon.com/sagemaker/pricing/
[6] [n. d.]. TFRecords Format details. https://www.tensorflow.org/api_guides/python/python_io#tfrecords_format_details
[7] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2009. *Above the Clouds: A Berkeley View of Cloud Computing*. Technical Report UCB/EECS-2009-28. EECS Department, University of California, Berkeley.
[8] AWSlabs. 2018. amazon-sagemaker-examples. https://github.com/awslabs/amazon-sagemaker-examples/tree/master/sagemaker-python-sdk/tensorflow_distributed_mnist.
[9] Google Cloud Platform. 2018. cloudml-dist-mnist-example. https://github.com/GoogleCloudPlatform/cloudml-dist-mnist-example.
[10] Daniel Pop. 2016. Machine Learning and Cloud Computing: Survey of Distributed and SaaS Solutions. *CoRR* abs/1603.08767 (2016). arXiv:1603.08767 http://arxiv.org/abs/1603.08767
[11] TensorFlow. 2017. cnn-mnist. https://github.com/tensorflow/tensorflow/blob/r1.8/tensorflow/examples/tutorials/layers/cnn_mnist.py.
[12] Blesson Varghese, Ozgur Akgun, Ian Miguel, Long Thai, and Adam Barker. 2016. Cloud Benchmarking For Maximising Performance of Scientific Applications. *CoRR* abs/1608.00406 (2016). arXiv:1608.00406 http://arxiv.org/abs/1608.00406